

# DEVICE AND METHOD FOR IMAGE PROCESSING

**Publication number:** JP11328389

**Publication date:** 1999-11-30

**Inventor:** KABURAGI HIROSHI

**Applicant:** CANON KK

**Classification:**

- international: **H04N1/407; G06T5/00; H04N1/407; G06T5/00; (IPC1-7): G06T5/00; H04N1/407**

- European:

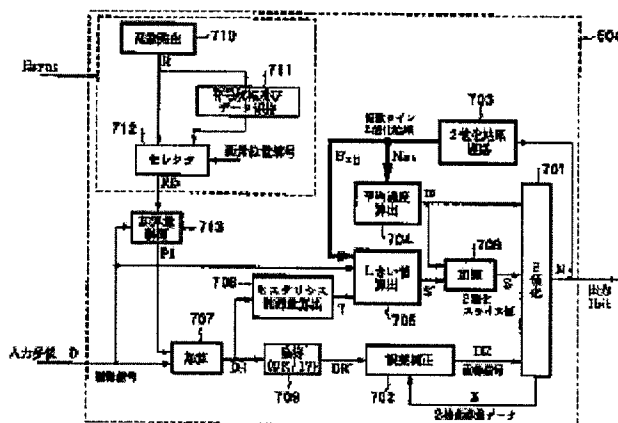
**Application number:** JP19980129158 19980512

**Priority number(s):** JP19980129158 19980512

Report a data error here

## Abstract of JP11328389

**PROBLEM TO BE SOLVED:** To simplify the hardware and to convert the gradations of multivalued image data with high picture quality by performing the gradation conversion by adding a random numbers value to the inputted multivalued image data, dividing the addition result by a specific value, and outputting only the quotient. **SOLUTION:** A sign inversion and data hold part 711 converts the sign of random numbers R inputted from a random numbers generation part 710 and outputs hold random numbers to a selector 712. The selector 712 switches and outputs the random numbers R inputted from the random numbers generation part 710 and the hold random numbers from the sign inversion and data hold part 711. An addition part 707 performs an adding process between an inputted image signal D and an addition quantity control part 713. A hysteresis controlled value calculation part 708 calculates the controlled value of hysteresis with the signal from the addition part 707 and outputs it to a threshold value calculation part 705. A division part 709 divides the inputted image signal DR by a constant 17 and outputs only its quotient. Thus, a gradation converting process part 604 performs binarizing processing.



Data supplied from the esp@cenet database - Worldwide

(51)Int.Cl.<sup>8</sup>

識別記号

F I

G 0 6 T 5/00

C 0 6 F 15/68

3 2 0 A

H 0 4 N 1/407

H 0 4 N 1/40

1 0 1 E

審査請求 未請求 請求項の数7 O L (全 14 頁)

(21)出願番号 特願平10-129158

(22)出願日 平成10年(1998)5月12日

(71)出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 燕木 浩

東京都大田区下丸子3丁目30番2号 キヤ  
ノン株式会社内

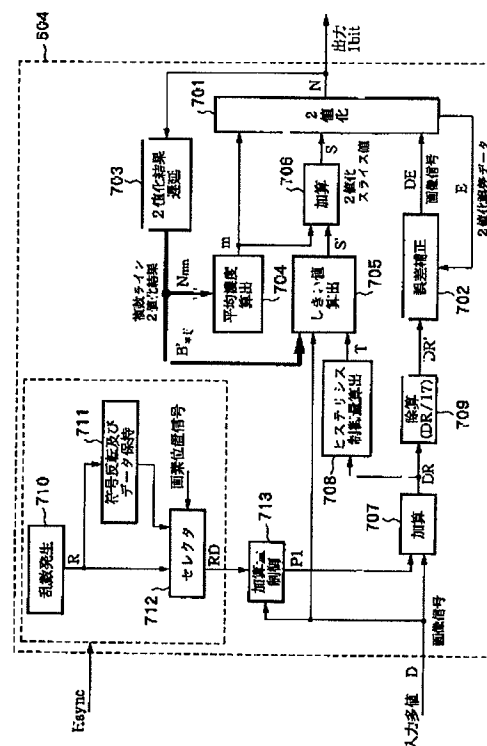
(74)代理人 弁理士 大塚 康徳 (外2名)

(54)【発明の名称】 画像処理装置及び方法

(57)【要約】

【課題】 ハードウェアの簡略化を図れると共に、多値画像データの階調変換処理を高画質に行える画像処理装置及び方法を提供する。

【解決手段】 乱数発生部710で乱数値を生成し、加算部707で多値の画像データに乱数値を加算し、除算部709で加算結果を所定の値で除算し、商のみを出力して入力した8ビットの画像データを4ビットに階調変換する。そして、平均濃度保存法により2値化して出力する。



## 【特許請求の範囲】

【請求項1】 入力した多値の画像データを階調変換して出力する画像処理装置において、乱数値を生成する乱数生成手段と、入力した多値の画像データに前記乱数値を加算する加算手段と、前記加算手段での加算結果を所定の値で除算し、商のみを出力して階調変換を行う階調変換手段とを有することを特徴とする画像処理装置。

【請求項2】 前記乱数生成手段は、±に変化する所定の振幅を有する乱数値を生成することを特徴とする請求項1記載の画像処理装置。

【請求項3】 前記加算手段は、前記前記所定の値による除算値の1/2をバイアス値として更に加算することを特徴とする請求項2記載の画像処理装置。

【請求項4】 前記乱数生成手段は、正の値を有する乱数値を生成することを特徴とする請求項1記載の画像処理装置。

【請求項5】 前記所定の値が偶数の場合、前記画像データ及び所定の値を2倍にして演算することを特徴とする請求項1記載の画像処理装置。

【請求項6】 更に、前記階調変換手段の出力値をt値化するt値化手段を有することを特徴とする請求項1記載の画像処理装置。

【請求項7】 入力した多値の画像データを階調変換して出力する画像処理方法において、乱数値を生成する乱数生成工程と、入力した多値の画像データに前記乱数値を加算する加算工程と、前記加算工程での加算結果を所定の値で除算し、商のみを出力して階調変換を行う階調変換工程とを有することを特徴とする画像処理方法。

## 【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、入力した多値の画像データを階調変換して出力する画像処理装置及び方法に関するものである。

【0002】

【従来の技術】従来、中間調表現を行うための画像処理方法として誤差拡散法（以下「ED」と呼ぶ）や平均濃度保存法（以下「MD」と呼ぶ）などが一般に知られている。これらは、少ない階調数を用いて面積階調表現を行うことにより、マクロ的に中間調を表現しようとするものである。

【0003】しかし、これらの処理は入力画像（8bit）に対して行うため、処理負荷が大きかった。そのため、入力画像の8bitを前処理で4bitに変換し、処理負荷を小さくする手法が米国特許USP5,394,250やUSP5,436,736などで提案されている。

【0004】

【発明が解決しようとする課題】しかしながら、上述の提案は、処理負荷を小さくすることができても、EDやMDで処理したときに特定の濃度領域で問題となる独自のテクスチャを改善することができなかった。

【0005】そこで、図1に示すように、t値化処理の前段で入力画像に所定の乱数を加算することにより、そのテクスチャを改善する手法が提案されている。その詳細を以下に述べる。

【0006】同図において、101は乱数発生部であり、乱数Rを生成する処理部である。図2は乱数生成部101の構成を示すブロック図である。また、図3は乱数発生をプログラム言語Cで示したものである。ここでは、説明の関係上、図3を用いて説明する。

【0007】まず、初期化で、 $p[ii] : (0 \leq ii \leq 25)$  のレジスタに“0”を書き込み、 $p[12]$  のレジスタのみに“1”を設定する。そして、乱数値を出力する前に、画素毎に  $p[0] = (p[25] \wedge p[24] \wedge p[23] \wedge p[22] \& 1)$  の演算を行った後、以下の演算により、-17～17の乱数値を生成する。

【0008】乱数 =  $(1 - 2 * p[22]) * ((p[15] * 64 + p[16] * 32 + p[17] * 16 + p[18] * 8 + p[19] * 4 + p[20] * 2 + p[21]) * 17) / 28$

ここで、±17以上の乱数を生成し、最後に128で除算しているのは、乱数の発生率に偏りをなくするためである。

【0009】また、上式では、-17～17の乱数を生成しているが、乱数発生部を以下のようにし、-15～15までの乱数値を生成するように変更しても同様な効果が得られることは言うまでもない。

【0010】乱数 =  $(1 - 2 * p[2]) * (p[18] * 8 + p[19] * 4 + p[20] * 2 + p[21])$

この乱数値は、乱数値の絶対値を大きくするとテクスチャ改善効果が大きくなり、絶対値を小さくするとテクスチャ改善効果が小さくなる傾向がある。

【0011】しかし、乱数値を大きくしすぎると、テクスチャの改善効果以上に画像が荒れてしまい、画質の低下につながることは言うまでもない。絶対値で7～17程度の値が実現的である。

【0012】図1に戻り、乱数発生部101から出力された乱数値は、符号反転及びデータ保持部102とセレクタ103に入力される。

【0013】符号反転及びデータ保持部102では、入力された乱数値の符号を一度反転し、“ $p \times X + p / 2$ ”（ $p \geq 2$ の任意な偶数値、X：主走査方向のアドレス値）画素の間、その乱数値を保持した後、出力する構成となっている。

【0014】一方、セレクタ103では、入力された乱数値と、符号反転及びデータ保持部102との値を画像位置信号により、出力を切り替える構成となっている。

【0015】ここで、上述の画素位置信号とは、“ $p \times X$ ” ( $p \geq 2$ の任意な偶数値、 $X$ :主走査方向のアドレス値)の画素毎に乱数発生部101の値を出力させ、

“ $p \times X + p/2$ ”の画素毎に符号反転及びデータ保持部102の値を出力させる制御信号のことであり、 $H_{sync}$ 信号から生成される。それ以外の画素位置では、“0”の値を出力する構成となっている。

【0016】以上の処理により生成した乱数値を、入力多値信号Dが“0”又は“255”以外の場合、加算部104で入力多値信号Dに加算する構成となっている。この時、図示はしていないが、最小0、最大255でリミットをかけている。

【0017】次に、除算部105では、入力画像信号を“17”で割り、商と余りとを分離する処理を行っている。ここで、商(0~15)は、加算部107へ入力され、余り(0~16)は、コンパレータ106へ入力される構成となっている。

【0018】また、乱数発生部109では、上述の乱数発生部101と同様に同様な手法で生成された0~16の正の乱数値が出力される構成となっている。その詳細は、乱数発生部101と同様なため、説明を省略するが、図4は乱数発生部109の乱数発生をプログラム言語Cで示したものである。

【0019】次に、コンパレータ106では、乱数発生部109から入力された乱数値(0~16)と除算部105から入力された余りの値(0~16)とを比較し、余りの方が乱数値より大きい場合の時のみ、“1”の値を出力し、それ以外は“0”を出力する構成となっている。この出力値は、加算部107で除算部105からの商の値(0~15)が加算され、4bit信号が後述するt値化処理部に出力される。

【0020】以上の処理により、入力信号を4bit信号に変換した後、t( $t \leq 4$ )値化処理をt値化処理部108で施し、出力する構成となっている。

【0021】しかし、上述した図1に示した手法の場合、加算部104と加算部107とで、入力画像に2度乱数を加えていることになり、ハード規模が大きくなるだけでなく、画質の低下にもつながるという問題点があった。

【0022】本発明は、上記課題を解決するためになされたもので、ハードウェアの簡略化を図れると共に、多値画像データの階調変換処理を高画質に行える画像処理装置及び方法を提供することを目的とする。

【0023】

【課題を解決するための手段】上記目的を達成するために、本発明は、入力した多値の画像データを階調変換して出力する画像処理装置において、乱数値を生成する乱数生成手段と、入力した多値の画像データに前記乱数値を加算する加算手段と、前記加算手段での加算結果を所定の値で除算し、商のみを出力して階調変換を行う階調

変換手段とを有することを特徴とする。

【0024】また上記目的を達成するために、本発明は、入力した多値の画像データを階調変換して出力する画像処理方法において、乱数値を生成する乱数生成工程と、入力した多値の画像データに前記乱数値を加算する加算工程と、前記加算工程での加算結果を所定の値で除算し、商のみを出力して階調変換を行う階調変換工程とを有することを特徴とする。

【0025】

【発明の実施の形態】以下、図面を参照しながら本発明に係る実施の形態について詳細に説明する。尚、本実施形態では、本発明をカラー複写機の2値化処理に適用した場合を例に説明する。

【0026】●処理概略

図5は、実施形態におけるカラー複写機の概略構成を示すブロック図である。同図において、509は画像読取部であり、レンズ501、CCDセンサ502、アナログ信号処理部503により構成される。ここで、レンズ501を介してCCDセンサ502に結像された原稿500の画像情報が、CCDセンサ502によりR(Red)、G(Green)、B(Blue)のアナログ電気信号に光電変換される。変換された画像信号は、アナログ信号処理部503に入力され、R、G、B、の各色毎にサンプル&ホールド、ダークレベルの補正等が行われた後にアナログ・デジタル(A/D)変換される。そして、デジタル変換されたフルカラー画像信号は、504の画像処理部に入力される。

【0027】画像処理部504では、シェーディング補正、色補正、γ補正等の画像読取系で必要な補正処理や、スムージング処理、エッジ強調、その他の処理、加工等が行われ、505のプリンタ部に出力される。

【0028】プリンタ部505は、レーザ等からなる露光制御部(図示せず)、画像形成部(図示せず)、転写紙の搬送制御部(図示せず)等により構成され、入力された画像信号により転写紙上に画像を記録する。

【0029】510はCPU回路部であり、CPU506、ROM507、RAM508等により構成され、画像読取部509、画像処理部504、プリンタ部505等を制御し、本装置のシーケンスを統括的に制御する。

【0030】●画像処理部

次に、上述の画像処理部504について詳細に説明する。図6は、実施形態における画像処理部504の構成を示すブロック図である。同図において、601はシェーディング補正部、602は階調補正部、603はカラー/モノクロ変換部、604は階調変換処理部である。

【0031】上記の構成において、まずアナログ信号処理部503より出力されるデジタル画像信号は、シェーディング補正部601に入力される。シェーディング補正部601では、原稿を読み取るセンサのばらつきや原稿照明用ランプの配光特性の補正を行っている。次に、

補正演算された画像信号は、輝度信号から濃度データに変換するために階調補正部602に輸入され、濃度画像データが作成される。濃度データに変換された画像信号は、カラー／モノクロ変換部603に輸入され、モノクロデータとして出力される。そして、カラー／モノクロ変換部603から出力されたデータは、階調変換処理部604に輸入され、擬似中間調表現として誤差拡散処理（ED処理）若しくは平均濃度保存処理（MD処理）が行われる。

【0032】次に、本発明に係る階調変換処理部604について詳細に説明する。

#### 【0033】●階調変換処理部

図7は、実施形態における階調変換処理部604の詳細な構成を示すブロック図である。尚、実施形態においては、テクスチャ制御を可能とした2値のMD法を例に挙げて説明する。

【0034】カラー／モノクロ変換部603からの画像信号Dに後述する乱数が加算され、更に後述する定数“17”による除算処理が行われた信号DR'と、2値化処理で発生した誤差データEとが誤差補正部702に輸入される。そして、後述する誤差補正を行い、画像信号DEとして2値化部701に出力する。

【0035】2値化部701は、誤差補正された画像信号DEと、後述する2値化スライス値Sと、後述する平均濃度値mとを輸入し、画像信号DEと2値化スライス値Sとを比較することによって2値出力Nを求める。また、画像信号DEと平均濃度値mとを減算処理することによって2値化誤差データEを算出する。

【0036】2値化結果遅延部703は、2値化された2値出力Nを輸入し、所定のライン数の遅延を行い、遅延されたデータを複数ライン2値化結果Nm<sub>n</sub>及びB\*<sub>i</sub><sub>j</sub>として平均濃度算出部704、及びしきい値算出部705に送出する。

【0037】平均濃度算出部704は、遅延された複数ライン2値化結果Nm<sub>n</sub>を輸入し、予め設定してある係数と積和演算を行って平均濃度値mを算出し、加算部706及び2値化部701へ平均濃度値mを出力する。

【0038】しきい値算出部705は、上述の2値化結果遅延部703の複数ライン2値化結果B\*<sub>i</sub><sub>j</sub>と、入力多値データDと、ヒステリシス制御量算出部708の出力Tとを輸入し、過去の2値化状況（パターン）であるB\*<sub>i</sub><sub>j</sub>信号に応じて任意の領域におけるしきい値制御量を算出して、それを2値化スライス値S'として加算部706へ送出する。

【0039】加算部706は、平均濃度算出部704の平均濃度値mと、しきい値算出部705の2値化スライス値S'との信号を輸入し、加算処理を行って、その結果を2値化スライス値として2値化部701へ出力する。

【0040】乱数発生部710は、後述する手法により

「-17～17」のm系列の乱数Rを発生させ、セクタ712と符号反転及びデータ保持部711へ出力する。

【0041】符号反転及びデータ保持部711は、乱数発生部710から入力した乱数Rの符号反転を行い、後述する一定画素間だけデータを保持した後、セクタ712に保持乱数を出力する。

【0042】セクタ712は、後述するタイミングの画素位置信号に基づき、乱数発生部710から入力した乱数Rと符号反転及びデータ保持部711からの保持乱数とを切り替えて出力する。

【0043】加算量制御部713は、画像信号Dの値に応じて、後述する手法を用いて乱数出力値の制御を行う。

【0044】加算部707は、入力された画像信号Dと加算量制御部713との加算処理を行う。ヒステリシス制御量算出部708は、加算部707からの信号によって後述する手法でヒステリシスの制御量を算出し、しきい値算出部705へ出力する。除算部709は、入力された画像信号DRを定数17で除算し、その商のみを出力する。この時、余りはすべて切り捨てている。

【0045】以上の構成により、階調変換処理部604で2値化処理が行われる。

【0046】次に、階調変換処理部604の各処理部について詳細に説明する。まず、乱数発生部710は、前述した図1と同様な構成で乱数が生成される。

【0047】まず、初期化で、p[i] : (0 ≤ i ≤ 25) のレジスタに“0”を書き込み、p[12]のレジスタのみに“1”を設定する。そして、乱数値を出力する前に、画素毎にp[0] = (p[25] ^ p[24] ^ p[23] ^ p[22] & 1) の演算を行った後、以下の演算により、-17～17の乱数値を生成する。

【0048】乱数 = (1 - 2 \* p[22]) \* ((p[15] \* 64 + p[16] \* 32 + p[17] \* 16 + p[18] \* 8 + p[19] \* 4 + p[20] \* 2 + p[21]) \* 17 / 128)

実施形態では、-17～17の乱数を用いているが、乱数発生部を以下の演算により、-15～15までの乱数値を生成するように変更してもよい。

【0049】乱数 = (1 - 2 \* p[2]) \* (p[18] \* 8 + p[19] \* 4 + p[20] \* 2 + p[21])

ここで、重要なことは、生成した乱数の最大値（実施形態では17）が、後述する除算部709で除算する数（実施形態では17）の1/2以上（少数部切り捨て）にする必要があることである。

【0050】尚、実施形態では、除算部706で除算する数が17であるため、17/2 = 8（少数部切り捨て）となり、乱数発生部710の最大乱数生成値は8以上の値に設定する必要があることになる（実施形態では8以上の17に設定している）。もし、除算部706で

除算する数が7であるならば、 $7/2=3$ （少数部切り捨て）となり、乱数発生手段310の最大乱数生成値は、3以上の値にする必要があることは言うまでもない。

【0051】このように、乱数生成を全ての画素に対して行う。

【0052】次に、乱数反転及びデータ保持部711では、“ $p \times X$ ”（ $p \geq 2$ の偶数、 $X$ ：主走査方向のアドレス値）の画素位置で生成した乱数発生部710の乱数値を、符号のみ反転して、“ $p/2$ ”画素の間保持した後、出力構成となっている。例えば、 $p$ の値が“2”の場合、“ $p \times X$ ”の画素位置、つまり、画素位置が“0, 2, 4, 6, 8, 10, 12, 14, …”で発生した乱数値を一時的に保持し、画素位置が“1, 3, 5, 7, 9, 11, 13, 15, …”で保持していた乱数値の符号を反転して出力することになる。無論、全ての画素の乱数値を保持し続ける必要はなく、一画素ずつリフレッシュしていく構成で十分である。

【0053】一方、セクタ712は、画素毎に生成される乱数発生部710の乱数値と、符号反転及びデータ保持部711とからの乱数値を、画素位置信号によって切り替えて出力する構成となっている。

【0054】この画素位置信号とは、“ $p \times X + p/2$ ”（ $p \geq 2$ の偶数、 $X$ ：主走査方向のアドレス値）の画素位置のときのみ、符号反転及びデータ保持部711からの乱数値をセレクトし、それ以外のときは、すべて、乱数発生部710からの乱数値をセレクトするものである。

【0055】加算量制御部713は、上述の乱数発生部710における量大乱数値が、後述する除算部709で割る数（実施形態では17）の $1/2$ （少数部切り捨てで8）より大きい場合に、その大きい乱数に対してのみ、入力多値信号Dに応じた出力制御を行う構成が重要なポイントとなっている。

【0056】図8は、加算量制御部713の加算量制御をプログラム言語Cで示したものである。ここで、重要なことは、セクタ712の出力RDを定数SLで除算する定数SLの設定法である。

【0057】定数SLは、 $RD/SL$ の結果が、除算部709で割る数の $1/2$ になるように決定してある。つまり、実施形態では、除算部709で割る数が17であるため、 $17/2=8$ （少数部切り捨て）となり、 $RD/SL=8$ 、 $17/SL=8$ （少数部切り捨て）からSL値を“2”に設定する必要がある。

【0058】入力多値信号DがN1（例えば16）以下の場合には、“ $P1=RD/SL$ ”の演算により、必要最小限の乱数が加算されることになる。ここでいう必要最小限の乱数とは、除算部709において“17”で除算するため、乱数の振れ幅を“ $-8 \sim 8$ ”に設定したということである。つまり、乱数の振れ幅を $\alpha$ （ここで

は、16）とすると、除算部709で割る数は、 $(\alpha + 1)$ （ここでは、17）となる関係になる。

【0059】また、入力多値信号DがN1より大きくN2（例えば32）以下の場合には、“ $P1=(RD-RD/SL) \times (D-N1)/(N2-N1)+RD/SL$ ”の演算により、振幅制御された乱数が加算されることになる。ここでのポイントは、必要最小限の乱数以上の部分に対してのみ、振幅制御されるということである。その部分が、上式の中の“ $(RD-RD/SL) \times (D-N1)/(N2-N1)$ ”である。

【0060】同様に、入力多値信号DがN3（例えば201）以上でN4（例えば233）より小さい場合には、“ $P1=(RD-RD/SL) \times (N4-D)/(N4-N3)+RD/SL$ ”の演算により、上述した処理と同様に振幅制御された乱数が加算されることになる。このとき、必要以上の乱数で振幅制御されている部分は同様に“ $(RD-RD/SL) \times (N4-D)/(N4-N3)$ ”である。

【0061】また、入力多値信号DがN4以上の場合には、“ $P1=RD/SL$ ”の演算により、必要最小限の乱数のみが加算される。更に、入力多値信号Dが上記の範囲外の場合は、入力された乱数RDすべてが加算量制御部713から乱数P1として出力される構成となっている。

【0062】加算部707は、上述した乱数P1と入力多値信号Dと定数8とを加算する処理を行う。定数8を加算するというのが、ここでも重要なポイントとなっている。これは、後述する除算部709で割る定数が“17”であるため、除算した余りが最大16になることから、加算する乱数の振幅を16以上の偶数にする必要があり、 $16/2$ の演算から定数8が求められている。これが、バイアス成分として加算されている。

【0063】無論、もし、除算部709で割る数が“5”の場合には、加算部707で加算する定数を“2”とすることになる。図示はしていないが、加算結果が“0”，“255”の範囲に入るようにリミットがかけられている。そして、この加算部707からの信号は、除算部709とヒステリシス制御量算出部708とに入力されている。

【0064】除算部709は、既に上述したように、定数17で除算する演算を行っている。このとき、出力する信号は除算を行った商のみであり、余りはすべて切り捨てていることを特徴としている。つまり、実施形態では、従来例で用いられていた“除算した余り”と“乱数”とを比較するコンパレータを必要とせず、除算処理の商のみで8bit信号化する処理が可能となっている。無論、4bit化された画質は、従来例以上であることは言うまでもない。この除算部709からの出力信号DR'は、後述する誤差補正部702に入力され、誤差補正処理が行われる。

【0065】次に、テクスチャ制御を行う2値化手法について説明する。上述したように、誤差補正部702は、画像信号DRと2値化誤差データEとを入力し、画像信号DR'に誤差補正を行った画像信号DEを算出し、2値化部701へと出力するもので、図9に示すように構成されている。

【0066】入力された2値化誤差データEは、除算回路901によって1/2にされる。その結果は2系統に分岐され、その一方は減算回路902に入力され、もう一方はラインバッファ903に入力される。減算回路902では、2値化誤差データEとE/2との差EB(=E-E/2)を算出し、加算回路904にその結果を入力する。この時、図示していないが、リミッタ処理により、2値化誤差データEの取り得る値を、“-6~+6”にしている。

【0067】加算回路904では、複数ビット(実施形態の場合は3ビット)1ライン分のラインバッファ903によって1ライン分遅延されたEAと減算回路からのEBとの和を算出し、加算回路905に出力する。加算回路905では、加算結果のEA+EBと、画像信号DR'との和を算出し、画像信号DEとして出力する。つまり、誤差補正部702では、図10に示すように、注目画素「\*」に対して1ライン上の画素Aを2値化したときの2値化誤差EAと、1画素前の画素Bを2値化したときの2値化誤差EBの値を注目画素の値に加算する処理を行う。

【0068】次に、2値化部701は、前述した画像信号DEと、後述する2値化スライス値Sと、後述する平均濃度算出値mとを入力し、これらと比較することにより、2値出力N及び2値化誤差データEを出力するもので、図11に示すように構成されている。

【0069】入力された画像信号DEは、2系統に分岐され、その一方は比較回路1101に入力され、もう一方は減算回路1102に入力される。比較回路1101では、画像信号DEと2値化スライス値Sとの値を比較し、以下のように2値出力Nを出力する。

【0070】 $DE > S$  の時は、 $N = 1$   
 $DE \leq S$  の時は、 $N = 0$

また、減算回路1102では、平均濃度算出値mから画像信号DEの値を差し引き、2値化誤差データEとして出力する。

【0071】 $E = m - DE$

この時、前述したように、図示していないが、Eの値が“-6~+6”の範囲に入るようにリミッタ処理が行われている。

【0072】次に、2値化結果遅延部703は、2値化部701からの2値出力Nを入力し、所定のライン数の遅延を行い、複数ライン2値化結果Nm n、 $B * i j$ として平均濃度算出部704及びしきい値算出部705にデータを送るもので、図12に示すように構成されてい

る。

【0073】まず、入力された2値出力Nは1ビット1ライン分のラインバッファ1201からラインバッファ1202へと送られていき、データがライン毎に遅延されていく。また同時に、1画素分の遅延回路からなる遅延1203から遅延1208によって次々と1画素分の遅延がなされる。そして、遅延1206の出力、遅延1207の出力をそれぞれN14、N15として出力する。

【0074】ラインバッファ1201によって1ライン分遅延がなされた2値化データは、遅延1209から遅延1214によって遅延され、遅延1209から遅延1213の出力がN21からN25として出力される。また、ラインバッファ1202によって更に1ライン分遅延がなされた2値化データは、遅延1215から遅延1220によって遅延され、遅延1215から遅延1219の出力がN31からN35として出力される。

【0075】同時に、遅延1206から遅延1208の各出力をそれぞれB10、B20、B30として出力する。また、ラインバッファ1201によって1ライン分遅延がなされた2値化データは、遅延された後、それぞれB32からB02、Bi12からBi32として出力する。更に、ラインバッファ1202によって更に1ライン分遅延がなされた2値化データは、遅延された後、それぞれB31からB01、Bi11からBi31として出力される。

【0076】つまり、平均濃度算出部704には、2次元の画像を2値化したデータが複数ライン、複数画素の遅延処理が施され、複数ライン2値化結果Nm nとして、図13に示すような状態で平均濃度算出部704に入力される。

【0077】次に、平均濃度算出部704は、複数ライン2値化結果Nm nを入力し、予め設定してある係数と遅延された2値結果とから積和演算を行い、2値化部701と加算部706とで使用するデータmの出力を行うもので、図14に示すように構成されている。

【0078】つまり、乗算回路1401では、2値化データN15と係数M15とを入力し、両者の乗算結果を出力する。また乗算回路1402では、2値化データN14と係数M14とを入力し、両者の乗算結果を出力する。以下同様に、上述の演算を乗算回路1403から乗算回路1412の各々の回路によって行い、それらの乗算結果を加算回路1413によって全て足し込む。その結果を平均濃度算出値mとして出力する。図15は、平均濃度算出の処理を行うときの係数の例を示す図である。

【0079】次に、ヒステリシス制御量算出部708は、入力信号DRに応じて定数ALF(=32)の値を変化させてS'信号として出力する。これは、任意の濃度領域で、ヒステリシス量を調整するためである。つま

り、これにより任意の濃度領域でテクスチャ制御が可能となっている。

【0080】図16は、ヒステリシス制御量の算出処理をプログラム言語Cで示したものである。入力された信号DRが定数LR1(=16)以下の場合、11を“0”に設定するような処理を行い、入力された信号DRが定数LR1より大きく、かつ定数LR2(=48)以下の場合には、11を次式により求める。

【0081】 $11 = (DR - LR1) * (ALF * 256 / (LR2 - LR1)) / 256$ ;  
この演算により、入力信号DRの値が定数LR1から定数LR2に増加するに従い、11の値は、徐々に0から定数ALF(=32)に近づくことになる。

【0082】一方、入力信号DRが定数LR2より大きく、かつ定数LR3(=233)以下の場合には、11を一定な定数ALFとして出力する。また、入力信号DRが定数LR3より大きく、かつ定数LR4(=255)以下の場合には、11を次式により求める。

【0083】 $11 = ALF - ((DR - LR3) * (ALF * 256 / (LR4 - LR3))) / 256$ ;  
これは、入力信号DRの値がLR3から定数LR4に増加するに従い、出力11が、徐々に定数ALFから0に近づくことを示している。また、入力信号DRがLR4より大きい場合には、11を0に設定するような処理が行われる。

【0084】以上の処理後、11から定数ALFm(=16)を減算したものが、出力信号Tとして出力される。この減算を行う目的は、ヒステリシス制御量算出部708の信号Tを負の値から正の値まで変化させるためである。これにより、ラチチュードが広い範囲で任意の濃度領域におけるテクスチャ制御が可能となる。

【0085】次に、しきい値算出部705について説明する。図17は、しきい値算出処理をプログラム言語Cで示したものである。

【0086】まず、しきい値算出部705は、入力されたヒステリシス制御量算出部708の信号Tの値を、それぞれ定数LT1(=2), LT2(=4), LT3(=8), LT4(=16)で割って、内部で用いる変数A(=T/LT1), B(=T/LT2), C(=T/LT3), D(=T/LT4)を求める。

【0087】次に、後述する手法で、2値化結果遅延部703からの出力B' \* i jの2値化結果配置状態(パターン)に応じて2値化スライス値S'の値を変数A, B, C, Dと定数で制御する。図18は、2値化結果配置状態(パターン)を示す図である。この例では、高速処理のために注目画素の1つ前の画素を参照していない。無論、十分高速なロジックが組める場合は、注目画素の1つ前の画素を参照しても問題無いことは言うまでもない。

【0088】次に、実際に2値化結果の配置(パターン)に応じて、2値化スライス値Sを制御する処理につ

いて説明する。

【0089】注目画素の周りの2値化状況が以下の場合には、2値化スライス値Sを強制的にmaxの定数15にして出力する。これは、強制的にドットを打ちにくくするためである。

【0090】

$B32 == 0 \& B22 == 1 \& B12 == 0 \& B21 == 0 \& B11 == 1 \& B01 == 0$  or  
 $Bi12 == 0 \& Bi22 == 1 \& Bi32 == 0 \& B01 == 0 \& Bi11 == 1 \& Bi21 == 0$   
また、注目画素の周りの2値化状況が以下で、かつ入力値データDが、31(0~255中の31)未満の場合にも、2値化スライス値Sを強制的にmaxの定数15にして出力する。これも、上記の条件のときに、強制的にドットを打ちにくくするためである。

【0091】

$B12 == 0 \& B02 == 0 \& Bi12 == 0 \& Bi22 == 0 \& Bi32 == 0 \& B11 == 0 \& B01 == 0 \& Bi11 == 1 \& Bi21 == 0 \& Bi31 == 0 \& B20 == 0$   
一方、上記の条件で、入力多値データDが31(0~255中の31)以上の場合には、2値化スライス値Sを平均濃度算出値mに設定して出力を行う。これは、過去の2値化結果が特定の配列(パターン)になった場合には、テクスチャ制御を行わないようにするためである。無論、ここで定数31は、決まった値ではなく、パラメータであり、48や64などの別な値にも設定可能である。

【0092】このとき、31の値を大きくすると、積極的にテクスチャ制御がかかりやすくなり、逆に、小さくするとテクスチャ制御がかかりにくくなることは言うまでもない。

【0093】注目画素の周りの2値化状況が、以下の場合には、2値化スライス値Sを平均濃度算出値mから、変数Aを減算した値( $S = m - A$ )に設定して出力する。

【0094】 $B02 == 0 \& Bi12 == 0 \& B11 == 0 \& B01 == 1 \& Bi11 == 1 \& Bi21 == 0 \& B20 == 0$

これは上記の条件のときに、強制的にドットを打ち易くするためである。このときも、注目画素直前の2値化結果は参照しないで処理をおこなっている。

【0095】同様に、各2値化結果のパターンに応じて、注目画素直前の結果を参照せずに2値化スライス値Sの値を内部変数A, B, C, Dと定数とを用いて2値化スライス値S'を制御していく。その結果、ヒステリシス制御量算出値Tが正の場合には、ドットが打たれやすい方に制御され、ヒステリシス制御量算出値Tが負の場合には、ドットが打たれにくい方に制御される。

【0096】以上のような処理を各画素に関して順次行っていくと、ヒステリシス制御量算出値Tの値に応じて任意の濃度領域で、かつ、2値化結果遅延部の出力値B' \* i jの値に応じて任意の形のテクスチャに制御が可能となる。

【0097】実施形態では、2×2の画素単位のテクス



チャになるような制御を行っている。これにより、プリンタの特性で一画素が安定しない領域で任意の数のドットを集めて安定化させた画像形成が可能となる。

【0098】このようにして求められた2値化スライス値 $S'$ は、平均濃度算出部704の出力 $m$ と共に、加算部706に入力されて加算処理が行われる。このとき、 $S'$ の信号が15のときには、2値化スライス値 $S$ を15として出力し、それ以外のときには、 $S = S' + m$ の演算を行って出力している。図19は、上述の演算をプログラム言語Cで示したものである。

【0099】上述の加算処理により、2値化スライス値 $S$ が求められた後、上述した2値化部701により2値化処理が行われ、その2値信号が階調変換処理部604から出力され、プリンタ部505でプリントアウトされるように構成されている。

【0100】[変形例]次に、前述した階調変換処理部の変形例について説明する。変形例は、図7に示した乱数発生部周辺の構成を簡略化したものである。

【0101】図20は、変形例における階調変換処理部の詳細な構成を示すブロック図である。図7に示す構成と同様なものには同一符号を付け説明を省略する。

【0102】図20において、2001は乱数発生部であり、0から16までの正の乱数を生成する。具体的な構成は、図2に示した構成と同様である。つまり、初期化で $p[ii] : (0 \leq ii \leq 25)$ のレジスタに“0”を書き込み、 $p[12]$ のレジスタのみに“1”を設定する。そして、乱数値を出力する前に、画素毎に $p[0] = (p[25] \wedge p[24] \wedge p[23] \wedge p[22] \& 1)$

の演算を行った後、以下の演算により、0~16の乱数値を生成している。

乱数 =  $((p[15] * 64 + p[16] * 32 + p[17] * 16 + p[18] * 8 + p[19] * 4 + p[20] * 2 + p[21] * 16) / 128)$

ここで、16以上の乱数を生成し、最後に128で除算しているのは、乱数の発生率に偏りをなくするためである。

【0103】また、2002は加算部であり、入力された乱数 $R$ と入力多値データとを加算する処理を行う。変形例の特徴は、図7に示した加算部707と異なり、除算部709で除算した最大余り値16の $1/2$ をバイアス成分として加算していないことである。これは、加算する乱数が正の値のみをとり、負の値をとらないため、バイアス成分が必要なくなっているからである。

【0104】加算部2002からの出力値は、図7に示した構成と同様に、除算部709で除算処理され、商の値のみが出力される構成となっている。これにより、従来例で行っていた除算した結果の余りと乱数とを比較するコンパレータが省略可能となり、ハードウェア規模の簡略化を実現している。

【0105】以降、前述した実施形態と同様な処理の2値化処理が行われ、その2値信号が階調変換処理部604から出力され、プリンタ部505でプリントアウトされるように構成されている。

【0106】この変形例におけるポイントは、除算部709で除算する数が17であるため、乱数発生部2001で生成する最大乱数値を16以上（除算部709で除算する数； $17-1$ ）に設定していることである。無論、入力多値データが“0”，“255”以外の時に加算する乱数の量を多くすると、テクスチャの改善効果が大きくなることは言うまでもないが、大きな値に設定すると画質も低下する傾向がある。よって、31以下に設定することが好ましい。

【0107】また、前述した実施形態では、符号反転及びデータ保持部711において±に規則正しく変化する乱数を生成し、入力多値データ $D$ に加算したが、本発明は、これに限定するものではなく、符号反転及びデータ保持部711を用いず無作為に±に変化する乱数発生部の乱数を加算しても同様な効果が得られることは言うまでもない。

【0108】これにより、ハードウェアの規模を小さくすることができる。但し、実施形態で示した±に規則正しく変化する乱数の方が、低周波成分が低くなるため、画質の劣化が少ない傾向にある。

【0109】更に、前述した実施形態において、必要最小限の乱数8（絶対値で〔除算部709の数値； $17/2$ 〕、演算結果の少数部は切り捨て）のみを加算するときは、加算量制御部713が省略できることも言うまでもない。

【0110】そして、前述した実施形態の除算部709で除算する数が偶数（例えば18）の場合は、入力多値データを1bitシフトにより、2倍にして演算する構成とすることができる。

【0111】これは、18で除算する場合、除算した余りが最大17となるため、加算する乱数の振幅を17以上の奇数にする必要があるのだが、この奇数の振幅を均等な±に振り分けることはできないからである（バイアス成分が $17/2 = 8.5$ となり割り切れないため）。言いかえると、加算部707でバイアス成分を8.5という数値に設定することができないためである。

【0112】よって、入力多値データを2倍にして処理を行うように構成しても良い。

【0113】入力多値データを2倍し、除算部709で除算する数も2倍の36にし（ $18 \times 2$ ）、加算部707のバイアス成分を17（〔除算部709の数値； $36/2-1$ 〕）に設定することにより、加算する乱数の振幅を均等な±17に分割することが可能となる。これにより、除算部709で除算する数が偶数の場合でも、前述した構成で実現できることになる。

【0114】前述した実施形態と変形例では、2値化に

ついて説明したが、本発明はこれに限定するものではない。つまり、4値化、8値化などの前処理にも適用ができると共に、前述した平均濃度保存法(MD法)に限定するものでもなく、一般的な誤差拡散法(ED法)にも適用できることは言うまでもない。

【0115】また、本発明は、白黒(単色)の処理のみに限らず、カラーの信号に対しても適用できることも言うまでもない。

【0116】以上説明したように、実施形態によれば、テクスチャを改善する乱数加算処理とt値化処理の前処理として乱数を用いた8ビット→4ビット変換処理とを一つに纏めることにより、乱数加算量を減らしながらも、従来と同様なテクスチャの改善を図ることができ、かつ、入力データ8ビットを4ビットに変換する前処理が可能となる。

【0117】また、ハードウェア規模を小さくでき、画質も向上させることができる。

【0118】尚、本発明は複数の機器(例えば、ホストコンピュータ、インタフェイス機器、リーダ、プリンタなど)から構成されるシステムに適用しても、一つの機器からなる装置(例えば、複写機、ファクシミリ装置など)に適用してもよい。

【0119】また、本発明の目的は前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システム或いは装置に供給し、そのシステム或いは装置のコンピュータ(CPU若しくはMPU)が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。

【0120】この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0121】プログラムコードを供給するための記憶媒体としては、例えばフロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0122】また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS(オペレーティングシステム)などが実際の処理の一部又は全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0123】更に、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わる

CPUなどが実際の処理の一部又は全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0124】

【発明の効果】以上説明したように、本発明によれば、ハードウェアの簡略化を図れると共に、多値画像データの階調変換処理を高画質に行うことが可能となる。

【0125】

【図面の簡単な説明】

【図1】t値化処理の前処理における従来例の構成を示す図である。

【図2】図1に示す乱数生成部101の構成を示すブロック図である。

【図3】乱数発生をプログラム言語Cで示したものである。

【図4】乱数発生部109の乱数発生をプログラム言語Cで示したものである。

【図5】実施形態におけるカラー複写機の概略構成を示すブロック図である。

【図6】図5に示す画像処理部504の構成を示すブロック図である。

【図7】図6に示す階調変換処理部604の詳細な構成を示すブロック図である。

【図8】加算量制御部713の加算量制御をプログラム言語Cで示したものである。

【図9】誤差補正部702の詳細な構成を示す図である。

【図10】誤差補正部702で2値化誤差を注目画素に加算する処理を示す図である。

【図11】2値化部701の詳細な構成を示す図である。

【図12】2値化結果遅延部703の詳細な構成を示す図である。

【図13】2値化結果遅延部703で用いるデータの構成を示す図である。

【図14】平均濃度算出部704の詳細な構成を示す図である。

【図15】平均濃度算出部704で用いる係数の構成を示す図である。

【図16】ヒステリシス制御量の算出処理をプログラム言語Cで示したものである。

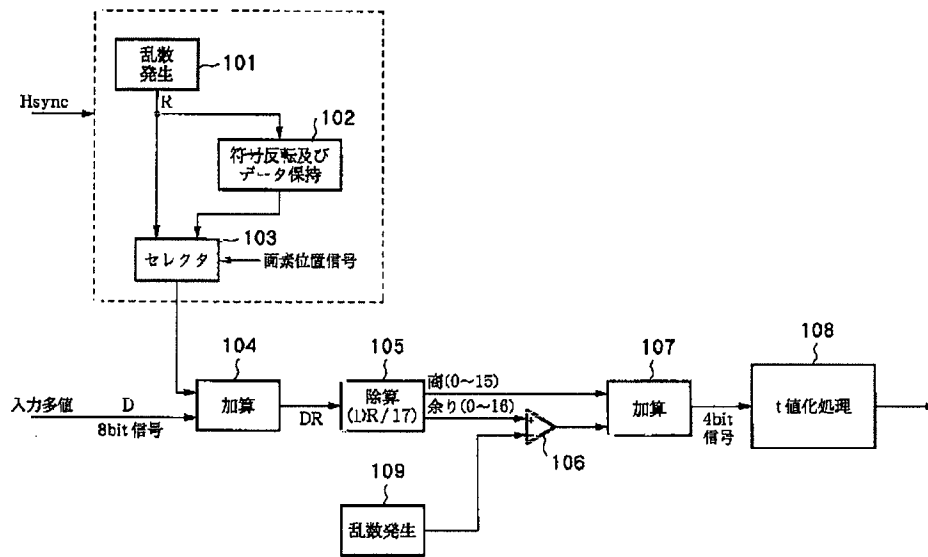
【図17】しきい値算出部705の処理をプログラム言語Cで示したものである。

【図18】2値化結果配置状態(パターン)を示す図である。

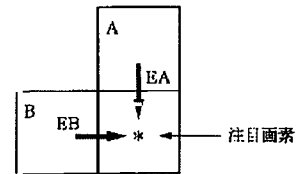
【図19】2値化スライス値Sの演算をプログラム言語Cで示したものである。

【図20】変形例における階調変換処理部の詳細な構成を示すブロック図である。

【図1】

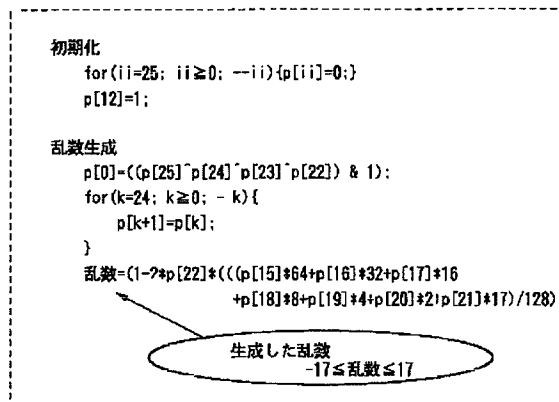
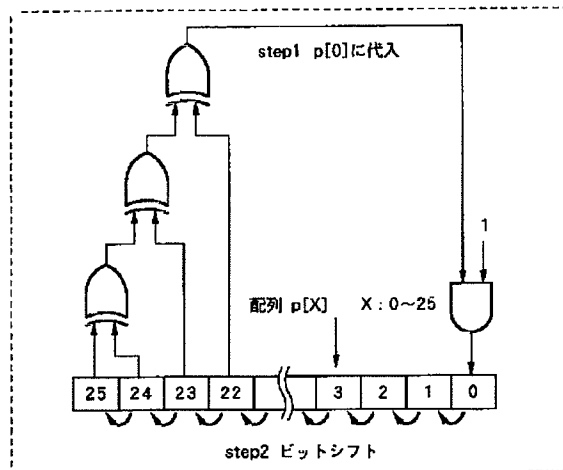


【図10】



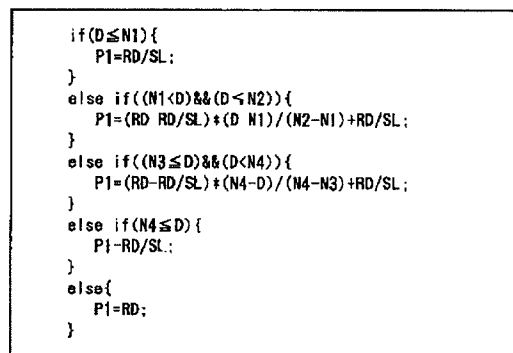
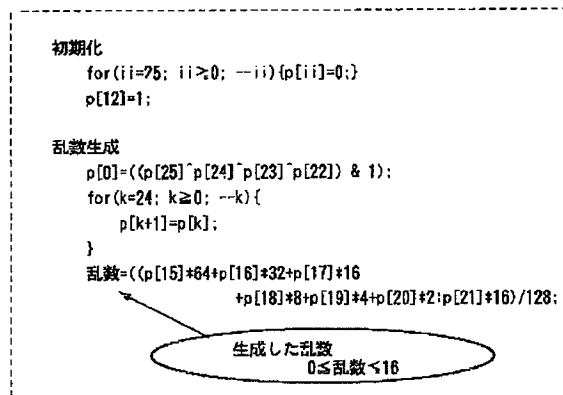
【図2】

【図3】



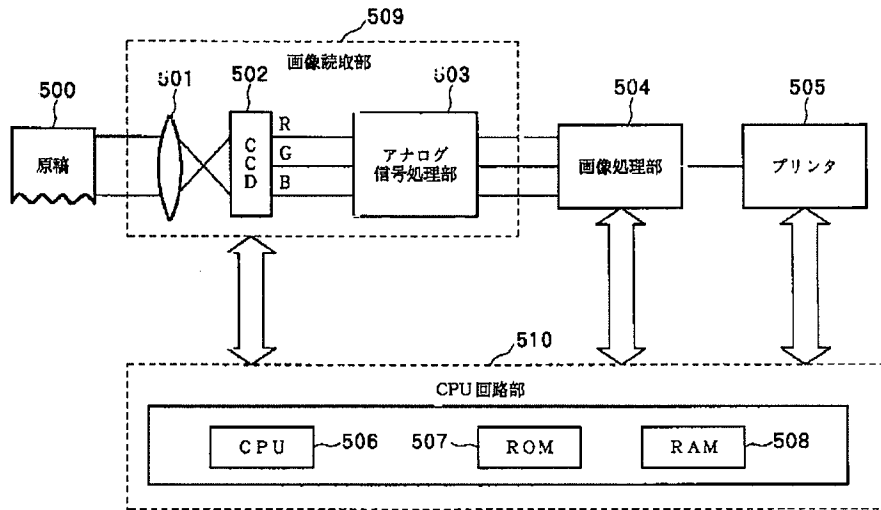
【図8】

【図4】

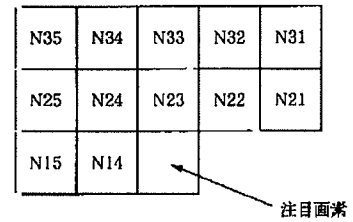


※除算演算結果はすべて切り捨て

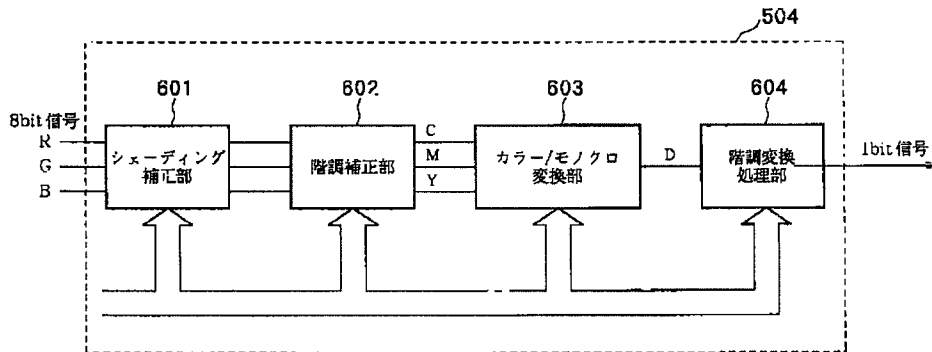
【図5】



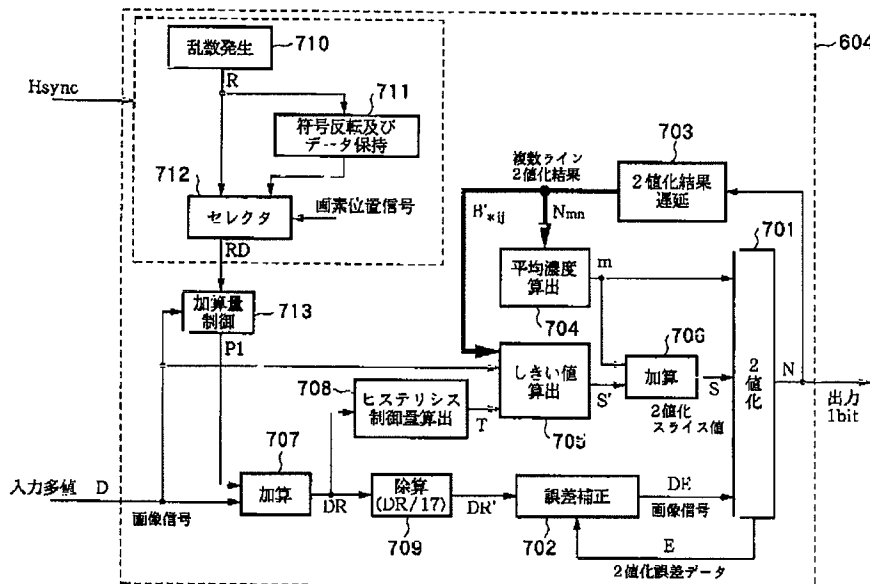
【図13】



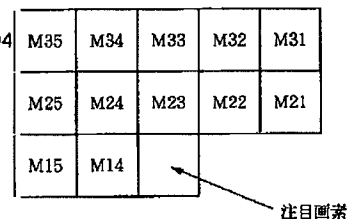
【図6】



【図7】



【図15】



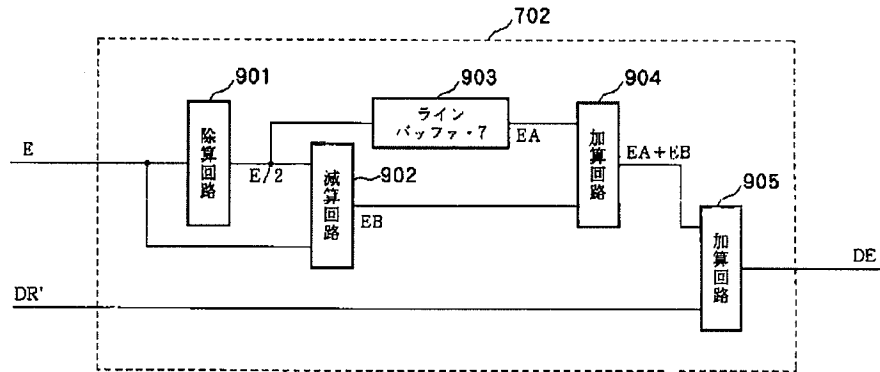
$$M35 \cdot M34 = M32 = M31 = M25 = 0$$

$$M33 = M21 = M15 = 1$$

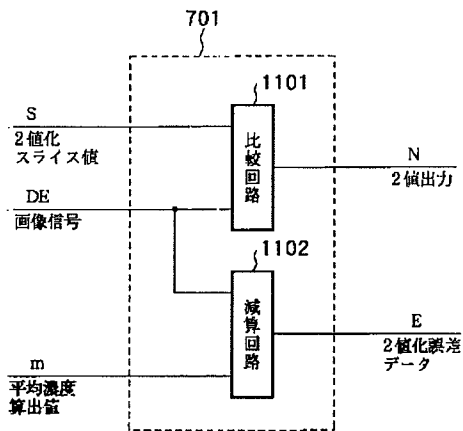
$$M24 \cdot M22 = 2$$

$$M23 \cdot M14 = 4$$

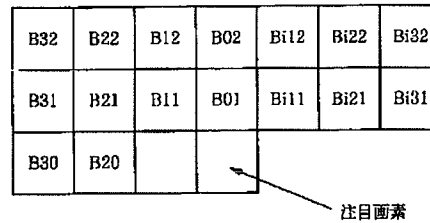
【図9】



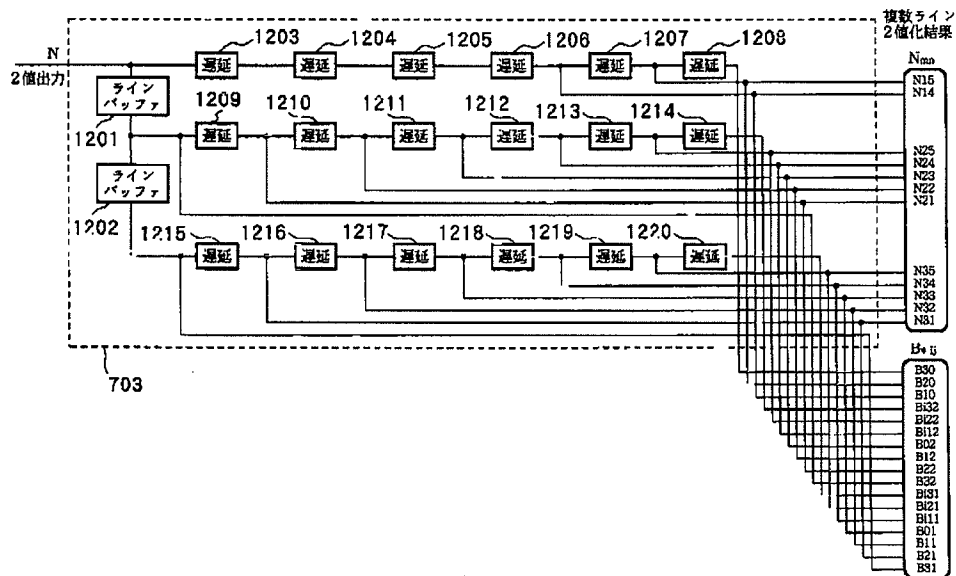
【図11】



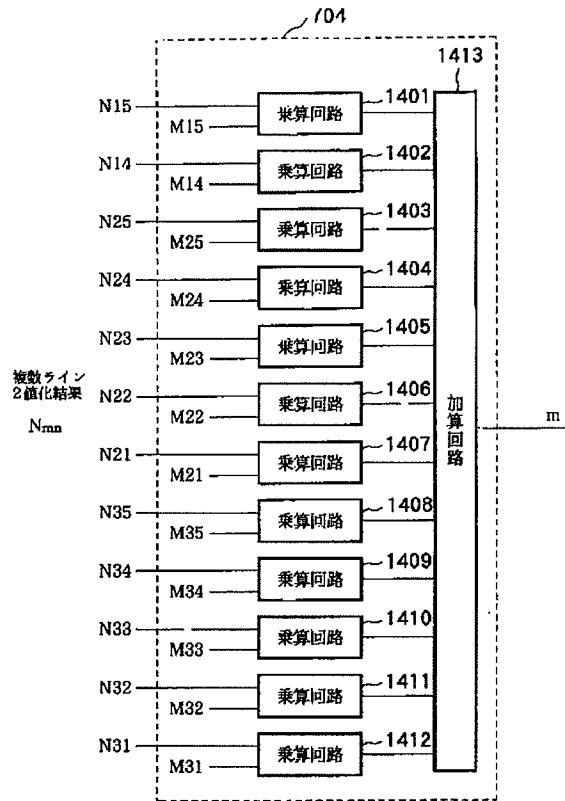
【図18】



【図12】



【図14】



【図17】

```

A-T/LT1;
B-T/LT2;
C-T/LT3;
D-T/LT4;

if(B32==0 && B22==1 && B12==0 && B21==0 && B11==1 && B01==0){
  S'=15;
}
else if(B12==0 && B22==1 && B32==0 && B01==0 && B11==1 && B21==0){
  S'=15;
}
else if(B12==0 && B02==0 && B12==0 && B22==0 && B32==0 && B11==0 && B01==0 && B11==1 && B21==0 && B31==0 && B20==0){
  if(D<31){S'=15;}
  else{ S'=0;}
}
else if(B32==0 && B22==0 && B12==0 && B02==0 && B12==0 && B31==0 && B21==1 && B01==0 && B11==0 && B20==0){
  if(D<31){S'=15;}
  else{ S'=0;}
}
else if(B02==0 && B12==0 && B11==0 && B01==1 && B11==1 && B21==0 && B20==0){
  S'=A;
}
else if((B02==0 || B12==0) && B11==0 && B01==1 && B11==1 && B21==0){
  S'=B;
}
else if(B12==0 && B02==0 && B21==0 && B11==1 && B01==1 && B11==0 && B20==0){
  S'=A;
}
else if((B12==0 || B02==0) && B21==0 && B11==1 && B01==1 && B11==0){
  S'=B;
}
else if(B12==0 && B02==0 && B21==0 && B11==0 && B21==0 && B20==0){
  S'=A;
}
else if(B12==0 && B02==1 && B12==0 && B21==0 && B11==1 && B01==0){
  S'=B;
}
else{
  S'=0;
}

```

※  
LT1 : 定数(2)  
LT2 : 定数(4)  
LT3 : 定数(8)  
LT4 : 定数(16)

【図16】

```

if(DR<=LR1){
  II=0;
}
else if(LR1<DR && DR<=LR2){
  II=((DR-LR1)*(ALF*256/(LR2-LR1)))/256;
}
else if(LR2<DR && DR<=LR3){
  II=ALF;
}
else if(LR3<DR && DR<=LR4){
  II=ALF-((DR-LR3)*(ALF*256/(LR4-LR3)))/256;
}
else{
  II=0;
}
T=II-ALFm

```

※  
LR1 : 定数(16)  
LR2 : 定数(48)  
LR3 : 定数(223)  
LR4 : 定数(255)  
ALF : 定数(32)  
ALFm : 定数(16)

【図19】

```

if(S'=15){S=S'; }
else{ S=S'+m; }

```

【図20】

